



TITLE:

多目的割当問題のミニマックス最適化(不確実性の下での意思決定と数理モデル)

AUTHOR(S):

ノムシリ, ウィーラユット; 山田, 武夫

CITATION:

ノムシリ, ウィーラユット ...[et al]. 多目的割当問題のミニマックス最適化(不確実性の下での意思決定と数理モデル). 数理解析研究所講究録 2006, 1477: 73-82

ISSUE DATE:

2006-03

URL:

<http://hdl.handle.net/2433/48266>

RIGHT:

多目的割当問題のミニマックス最適化

防衛大学校情報工学科 ウィーラユット・ノムシリ (VEERAYUTH Nomsiri)

防衛大学校情報工学科 山田 武夫 * (YAMADA Takeo)

Department of Computer Science,
The National Defense Academy

要旨: We are concerned with a variation of the assignment problem, where the assignment costs differ under different scenarios. We give a surrogate relaxation approach to derive a lower bound as well as an upper bound quickly, and show that the pegging test known for a 0-1 programming problem is also applicable to this problem. Next, we discuss how the computation time for pegging can be drastically shortened by taking the special structure of the assignment into account. Finally, through numerical experiments we show that the developed method finds upper and lower bound of high accuracy in relatively small CPU time, and also solve larger instances to optimality faster than conventional methods.

Keywords : assignment problem, robust optimization, pegging test, combinatorial optimization

1. はじめに

割当問題 (AP: assignment problem)[1] は与えられた n 人の学生 $I = \{1, \dots, n\}$ を n 個の仕事 $J = \{1, \dots, n\}$ に割り当てるとき, 割り当てにともなうコストの総和を最小化する問題である. AP の解法としては, ハンガリー法 [9, 11] が良く知られているが, スケーリング技法などを取り入れた, 多項式時間や強多項式時間のアルゴリズムも開発されている.

本研究では, これに対してシナリオが K 個あり [7], シナリオ k のときに, 学生 i を仕事 j に割り当てるコストが c_{ij}^k であるとする. 割り当て $x = (x_{ij})$ による総コストは, シナリオが k であれば

$$z^k(x) := \sum_{i=1}^n \sum_{j=1}^n c_{ij}^k x_{ij} \quad (1)$$

であるが, 本稿では, これらの最大のものを最小化する, 多目的割当問題のミニマックス最適化 (MMAP: mini-max assignment problem)[18] を考える.

MMAP:

$$\text{minimize} \quad \max_{1 \leq k \leq K} \left\{ \sum_{i=1}^n \sum_{j=1}^n c_{ij}^k x_{ij} \right\} \quad (2)$$

$$\text{subject to} \quad \sum_{j=1}^n x_{ij} = 1, \quad i = 1, 2, \dots, n \quad (3)$$

$$\sum_{i=1}^n x_{ij} = 1, \quad j = 1, 2, \dots, n \quad (4)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \quad (5)$$

*yamada@nda.ac.jp

MMAP は区分的に線形な凸関数を目的関数とする組合せ最適化問題で、目的関数を v と置くと、下のようにも書くことができる。

MMAP' :

$$\text{minimize} \quad v \quad (6)$$

$$\text{subject to} \quad \sum_{i=1}^n \sum_{j=1}^n c_{ij}^k x_{ij} \leq v, \quad k = 1, 2, \dots, K \quad (7)$$

$$(3), (4), (5)$$

この問題は、線形の混合整数計画問題であるので、ある程度のサイズの問題までは NUOPT や XPRESS-MP[13] などの商用ソフトで解くことができる。AP と異なり、MMAP は \mathcal{NP} -困難 [6] である。

AP は人員割当、機械スケジューリング問題のような様々な状況で応用される。MMAP は Kouvelis-Yu[7] によって定式化され、 \mathcal{NP} -困難であることが証明された。そして、小さい問題を解決した ($n \leq 40, K \leq 30$)。榊原ら [18] はシナリオ数 K が 2 で、 n の値が 200 までの問題について近似解を求める方法を発表している。Cimen[3] はトルコの軍隊の人員割当に MMAP を応用した。

本研究では、代理制約緩和と近似解法により良い上下界値が得られ、これに釘付けテストを適用すると問題を (大幅に) 縮小できることを示す。さらに、MMAP に対する釘付けテストの効果を高めるため **改善釘付けテスト** を新たに考案する。この方法によって通常の釘付けテストを用いたときに比べ、計算時間が大幅に短縮される。その結果、 $n = 1000$ までのサイズの問題を厳密に解くことが可能となる。

2. 上界値と下界値

2.1 代理制約緩和

$\lambda = (\lambda_1, \lambda_2, \dots, \lambda_K)$ を

$$\sum_{k=1}^K \lambda_k = 1, \quad \lambda_k \geq 0 \quad (k = 1, 2, \dots, K) \quad (8)$$

を満たす任意のベクトルとし、

$$\bar{c}_{ij}(\lambda) = \sum_{k=1}^K \lambda_k c_{ij}^k \quad (9)$$

と置くと、MMAP の代理制約緩和により下を得る。

SMMAP(λ):

$$\text{minimize} \quad \sum_{i=1}^n \sum_{j=1}^n \bar{c}_{ij}(\lambda) x_{ij} \quad (10)$$

$$\text{subject to} \quad (3), (4), (5)$$

$\lambda \in R_+^K$ を固定すると、SMMAP(λ) は通常の割当問題でハンガリー法などにより効率的に解くことができる。その最適解を $\underline{x}(\lambda)$ 、最適目的関数値を $\underline{z}(\lambda)$ とすると、 $\underline{z}(\lambda)$ は任意の

$\lambda \geq 0$ で MMAP の下界値を与える。また、 $z(\lambda)$ は λ に関して区分的に線形な凹関数となり、劣勾配法によって $z(\lambda)$ を最大とする λ^\dagger と $z := z(\lambda^\dagger)$ が得られる。以下、 z を最良の下界値とすると、次が示される。

命題 1

- (i) $\lambda \in R_+^K$ を固定すると、 $z(\lambda)$ は MMAP の一つの下界値を与える。MMAP の最適目的関数値を z^* とすると、これらの間には下の関係が成立する。

$$z^* \geq z(\lambda) \quad (11)$$

- (ii) $z(\lambda)$ は、 λ に関して区分的に線形な凹関数である。

- (iii) λ において、 $z(\lambda)$ が微分可能なら

$$\frac{\partial z(\lambda)}{\partial \lambda_k} = \sum_{i=1}^n \sum_{j=1}^n c_{ij}^k x_{ij} = z^k(\lambda) \quad (12)$$

任意の λ に対し、前に述べたように $z(\lambda)$ は MMAP の下界値を与えるが、この値はできるだけ大きい方が望ましい。このために、劣勾配法(subgradient method)[12, 17]を導入する。ここで劣勾配とは、(12)を成分とするベクトル $g := \partial z(\lambda)/\partial \lambda = (z^k(\lambda))$ を意味する。

また、ここでいう劣勾配法とは、目的関数 $z(\lambda)$ が微分不可能な点を有するという点を除いて通常の勾配法と同じで、次のように与えられる。

劣勾配法

- Step 1. $\lambda = (1/K, 1/K, \dots, 1/K)$ とする。
- Step 2. SMMAP (λ) を解く。
- Step 3. 劣勾配を計算し、探索方向 d を劣勾配 g から適当に定める。
- Step 4. (1 次元探索)
 $z(\lambda + \alpha d)$ が最大となる $\alpha^\dagger \geq 0$ を求める。
- Step 5. $\alpha^\dagger \cong 0$ の場合終了。
- Step 6. $\lambda \leftarrow \lambda + \alpha^\dagger d$ として Step 2 へ戻る。

ここで、 $\lambda + \alpha d$ が条件 (8) を満足するために、 d としては g を超平面

$$\lambda_1 + \lambda_2 + \dots + \lambda_K = 1$$

に射影して

$$d := g - \bar{g} \mathbf{1}$$

をとる。ここに

$$\bar{g} = \sum_{k=1}^K g_k / K$$

である。これが実際に $z(\lambda)$ の増加方向であることは、次のように証明できる。 $K=2$ の場合は、劣勾配法ではなく、2 分探索法を用いる。

2.2 上界値

代理制約緩和において、最適な λ^\dagger を求めるための劣勾配法 (あるいは 2 分探索法) において、様々な λ で $\text{SMMAP}(\lambda)$ を繰り返し解いたが、 $\text{SMMAP}(\lambda)$ の最適解 $\underline{x}(\lambda)$ は MMAP の実行可能解でもあるので、目的関数値 $\max_{1 \leq k \leq K} \{z^k(\underline{x}(\lambda))\}$ は z^* の上界値を与える。これを $\bar{z}(\lambda)$ 書き、 λ^\dagger を得るまでに得られた $\bar{z}(\lambda)$ のうち、最小のものを MMAP の上界値 \bar{z} とする。

3. 釘付けテスト

3.1 0-1 計画問題の釘付けテスト

$\text{SMMAP}(\lambda^\dagger)$ は標準的な割当問題なので、制約条件 (5) は非負条件 $x_{ij} \geq 0$ と改めてよい。この線形計画問題を解いた時の最適字引きを

$$z = \underline{z} + \sum_{j \in N} \alpha_{0j} y_j, \quad (13)$$

$$\bar{b}_i = y_{B(i)} + \sum_{j \in N} \alpha_{ij} y_j, \quad i = 1, 2, \dots, 2n-1 \quad (14)$$

とすると、最適性の定義より

$$\alpha_{0j} \geq 0, \quad \forall j \in N, \quad (15)$$

$$0 \leq \bar{b}_i \leq 1, \quad i = 1, 2, \dots, m+n. \quad (16)$$

ここに、 N は非基底変数の添字集合で、 $B(i)$ は i 番目の基底変数の添字を表す。また、最適性の条件より (15) である。

$$PU_s := \min \left\{ \frac{-\alpha_{0j}}{\alpha_{i(s)j}} \mid j \in N, \alpha_{i(s)j} < 0 \right\} (1 - \bar{b}_{i(s)}) \quad (17)$$

$$PL_s := \min \left\{ \frac{\alpha_{0j}}{\alpha_{i(s)j}} \mid j \in N, \alpha_{i(s)j} > 0 \right\} (\bar{b}_{i(s)}) \quad (18)$$

と置くと、最適解 (x_j^*) において次が成立する [10].

定理 1

P の最適解 $x^* = (x_j^*)$ において

(i) x_s が非基底変数の場合

$$\alpha_{0s} > \bar{z} - \underline{z} \quad \text{であれば, } x_s^* = 0 \quad (19)$$

(ii) x_s が基底変数の場合

$$PU_s > \bar{z} - \underline{z} \quad \text{であれば, } x_s^* = 0 \quad (20)$$

$$PL_s > \bar{z} - \underline{z} \quad \text{であれば, } x_s^* = 1 \quad (21)$$

3.2 MMAP の釘付けテスト

割当問題 (2) ~ (5) は 0-1 計画問題であるが、(5) を連続緩和して

$$x_{ij} \geq 0, \quad \forall i, j$$

と置きかえても良いことが知られている [1]. これは普通の線形計画問題であるので、市販の、あるいはフリーの線形計画ソフト [5] を用いて解くことができる. しかし、割当問題についてはその特殊構造を利用した、より効率的な解法がいくつか提案されている. 本節では、その中から最も基本的なハンガリー法 [9] について述べる. 上の線形計画問題の双対問題は

DAP:

$$\begin{aligned} & \text{maximize} && \sum_{i=1}^n p_i + \sum_{j=1}^n q_j && (22) \\ & \text{subject to} && p_i + q_j \leq c_{ij}, \quad \forall i, j && (23) \end{aligned}$$

となる. ベクトルの対 $p = (p_i)$, $q = (q_j)$ は、(23) を満たすとき、双対可能解という. ここで、節点集合 $V_1 = \{1, 2, \dots, n\}$, $V_2 = \{1', 2', \dots, n'\}$ と、等号が成り立つ枝だけからなる集合

$$A(p, q) = \{(i, j') \mid p_i + q_j = c_{ij}\}$$

を持つ 2 部グラフを $H(p, q)$ とする. このとき、線形計画法の双対性から $H(p, q)$ が完全マッチングを持つとき、そのマッチングが割当問題の解となる [2].

2 部グラフ $H(p, q)$ の左側節点の部分集合 $U \subseteq V_1$ が縮小であるというのは、 $|U| > |N(U)|$ であることをいう. ただし、 $N(U)$ は V_2 の節点で、 U に接するものの集合である.

次の定理は結婚定理、または Hall の定理として知られている [1].

定理 2

$H(p, q)$ が完全マッチングを持つことの必要十分条件は $H(p, q)$ は縮小を含まないことである.

ハンガリー法では、適当な双対可能解 (p, q) から出発し、グラフ $H(p, q)$ 中に縮小があるかどうか調べ、それがあつた場合には、(22) が増加するように (p, q) を修正する. $H(p, q)$ に縮小が含まれないようになるまでこれを反復することにより、割当問題の解が得られる.

ところで、SMMAP(λ^1) に前節の釘付けテストを適用するには、最適マッチング M に対応する最適単体表が必要である. これは、ハンガリー法の結果から次のように復元することができる. まず、次の事実に注意する.

命題 2

(i) $H(p, q)$ が連結グラフである場合、最適マッチング M を含む $H(p, q)$ の全域木 [8] が存在する.

(ii) 上の全域木に含まれる変数は SMMAP(λ^1) の最適基底を与える.

$H(p, q)$ が連結グラフでない場合には、 p, q を修正して $H(p, q)$ が連結となるようにできる. このとき、 $H(p, q)$ の全域木に対応する変数の集合 B が最適単体表の基底変数の集合とする. 非基底変数の集合は N と記す.

3.3 改善釘付けテスト

最初の単体表を

$$\begin{aligned} Bx_B + Nx_N &= \mathbf{1} \\ c_Bx_B + c_Nx_N &= z \end{aligned}$$

と記し、最適解において対応する単体表が

$$\begin{aligned} x_B + \bar{A}x_N &= \bar{b} \\ \bar{c}x_N &= z - \bar{z} \end{aligned}$$

であるとする。この場合

$$\begin{aligned} \bar{A} &= B^{-1}N, \quad \bar{b} = B^{-1}\mathbf{1} \\ \bar{c} &= c_N - c_BB^{-1}N, \quad \bar{z} = c_BB^{-1}\mathbf{1} \end{aligned}$$

で、 \bar{A} , \bar{c} の各成分が α_{ij} , α_{0j} に相当する。

ここで、定理 1 の (ii) を適用するには、 \bar{A} を求める必要があるが、 \bar{A} の大きさは $(2n-1) \times (n^2 - 2n + 1)$ で、 n が大きい場合、この要素をすべて計算するのは非常に負担が大きい。しかし、割当問題の特殊性から、この部分を次のように改善することができる。

連続緩和問題の最適解に対応する基底形式を考える。SMMAP(λ^+) の係数行列のユニモジュラー性から最適単体表では下が成立する。

$$\alpha_{ij} \in \{-1, 0, 1\}, \quad \forall j \in N \quad (24)$$

$$\bar{b}_i \in \{0, 1\} \quad (25)$$

そこで、

$$N^+ := \{j \in N \mid \alpha_{0j} > \bar{z} - \underline{z}\} \quad (26)$$

$$N^- := \{j \in N \mid \alpha_{0j} \leq \bar{z} - \underline{z}\} \quad (27)$$

とすると、次の定理を示すことができる。

定理 3

(i) $\bar{b}_i = 1$ の場合

$$PL_i > \bar{z} - \underline{z} \iff \{j \in N^- \mid \alpha_{ij} = 1\} = \emptyset$$

(ii) $\bar{b}_i = 0$ の場合

$$PU_i > \bar{z} - \underline{z} \iff \{j \in N^- \mid \alpha_{ij} = -1\} = \emptyset$$

証明: (i) (24) より、 $PL_i = \min\{\alpha_{0j} \mid \alpha_{ij} = 1, j \in N\} = \min\{PL_i^+, PL_i^-\}$ 。ただし、 $PL_i^+ = \min\{\alpha_{0j} \mid \alpha_{ij} = 1, j \in N^+\}$ と $PL_i^- = \min\{\alpha_{0j} \mid \alpha_{ij} = 1, j \in N^-\}$ である。(26) より、 $PL_i^+ > \bar{z} - \underline{z}$ であるから、明らかに $PL_i > \bar{z} - \underline{z} \iff \{j \in N^- \mid \alpha_{ij} = 1\} = \emptyset$ 。このようにして (i) が証明された。(ii) も同様。 ■

この定理の重要な含意は以下の事実である。すなわち、釘付けテストを実行する際に必要となるのは非基底変数 x_N に対応する行列 \bar{A} のすべてではなく、 N^- に含まれる列のみで、この N^- 部分について、 $\{j \in N^- \mid \alpha_{ij} = \pm 1\} = \emptyset$ が成り立てば、基底変数を 0 または 1 に固定することができる。

以上により, $\bar{A} = (\alpha_{ij})$ の $(n^2 - 2n + 1)$ 列すべてでなく, $\alpha_{0j} > \bar{z} - \underline{z}$ である列のみ計算すれば良いが, 通常このような列は非基底変数全体比べ, はるかに少数なので, 釘付け時間は大幅に短縮される.

4. 数値実験

4.1 実験計画

シナリオ数 K を 2 ~ 16, 商品数 n を 200 ~ 1000 の範囲で実験を行う. 最初に, 基準コスト c_{ij}^0 を $[1, 1000]$ の間に一様独立な乱数で発生し, δ を $0 < \delta < 1$ である定数として, シナリオ k のときの割当コストを次のように決める.

$$c_{ij}^k : [(1 - \delta)c_{ij}^0, (1 + \delta)c_{ij}^0] \text{ の整数一様乱数}$$

δ は割当コストのシナリオ間での相関の度合いを制御するパラメータで, δ が小さい程相関が強い. 以下では $\delta = 0.3$, $\delta = 0.6$, $\delta = 0.9$ の場合を考える.

上下界値と釘付けを計算するプログラムは C 言語で作成し, 実験は IBM RS/6000 SP44 MODEL270 (CPU : POWER 3-II SMP 2 WAY, 375MHz) 上で行う.

4.2 上下界値と釘付け結果

表 1, 表 2, 表 3 はそれぞれの δ ごとの実験結果を示す. ここで, \underline{z} と \bar{z} は下界値と上界値を表し, これらのギャップ ($= \bar{z} - \underline{z}$) は gap の欄に表している. #assign は λ^* が得られるまでに割当問題 SMMAP(λ) を解いた回数である. fix₁ は 1 に固定された変数の数を表しており, n' は固定されずに残った変数, つまり未定の変数の数を示している. CPU₁ は λ^* と上下界値の計算時間, CPU₂ は単体表の復元と釘付けテストに要した時間である. また, すべての行はランダムに発生した 10 例題についての平均値である.

表 1: 上下界値と釘付け ($\delta = 0.3$)

K	n	\underline{z}	\bar{z}	gap	#assign	fix ₁	n'	CPU ₁	CPU ₂
2	200	1459.1	1464.0	4.9	6.1	89.8	355.8	0.1	0.2
	600	1328.4	1330.6	2.2	5.9	167.3	1427.8	1.4	2.1
	1000	1132.9	1135.3	2.4	5.4	275.4	3627.9	4.0	10.2
4	200	1483.9	1490.0	6.1	44.2	62.3	409.5	0.9	0.2
	600	1340.1	1343.1	3.0	41.1	166.1	1563.9	11.0	2.1
	1000	1146.7	1149.1	2.4	33.3	294.3	2968.7	27.5	4.1
8	200	1487.0	1496.6	9.6	51.2	35.4	593.4	1.5	0.2
	600	1351.6	1357.9	6.3	38.1	31.3	2992.4	13.9	1.9
	1000	1155.6	1159.3	3.7	35.1	73.8	4763.6	38.9	11.5
16	200	1505.4	1516.9	11.5	58.1	25.3	698.4	2.9	0.2
	600	1354.8	1360.7	5.9	64.6	29.2	2876.0	35.9	2.0
	1000	1160.5	1164.9	4.4	52.0	47.2	5440.9	83.9	8.4

表 2: 上下界値と釘付け ($\delta = 0.6$)

K	n	\underline{z}	\bar{z}	gap	#assign	fix ₁	n'	CPU ₁	CPU ₂
2	200	1384.2	1393.2	9.0	6.1	63.2	581.8	0.1	0.2
	600	1252.9	1255.5	2.6	6.5	301.6	1381.9	1.6	0.9
	1000	1036.5	1039.8	3.3	6.2	332.8	4546.5	4.7	5.2
4	200	1449.7	1462.6	12.9	32.0	20.9	782.1	0.6	0.2
	600	1314.4	1322.3	7.9	37.0	31.4	3689.0	9.8	2.0
	1000	1113.1	1117.7	4.6	36.3	59.7	5658.7	30.9	10.2
8	200	1484.9	1506.9	22.0	50.4	8.4	1202.3	1.5	0.2
	600	1349.4	1364.0	14.6	49.9	9.7	6187.7	18.1	2.3
	1000	1145.7	1154.0	8.3	51.3	3.2	9706.3	56.0	11.5
16	200	1503.8	1530.2	26.4	58.2	2.7	1398.4	2.9	0.2
	600	1373.2	1388.2	15.0	64.8	1.2	6329.1	36.1	1.9
	1000	1167.6	1178.5	10.9	65.7	0.7	12428.9	108.0	9.8

表 3: 上下界値と釘付け ($\delta = 0.9$)

K	n	\underline{z}	\bar{z}	gap	#assign	fix ₁	n'	CPU ₁	CPU ₂
2	200	1195.8	1204.9	9.1	7.5	47.8	679.4	0.1	0.2
	600	989.7	999.4	9.7	6.9	48.5	5207.0	1.6	2.5
	1000	789.0	793.3	4.3	7.4	229.1	6445.3	5.4	10.2
4	200	1353.7	1383.0	29.3	51.6	3.9	1597.6	1.0	0.2
	600	1191.9	1202.5	10.6	35.7	17.1	4899.3	9.3	2.2
	1000	999.1	1010.0	10.9	31.2	3.3	13066.7	25.7	17.0
8	200	1433.0	1468.3	35.3	53.8	0.3	1804.8	1.6	0.2
	600	1279.1	1301.4	22.3	45.4	0.0	9144.6	16.4	2.6
	1000	1091.7	1104.1	12.4	54.3	0.0	14171.0	60.9	8.8
16	200	1485.2	1531.8	46.6	61.4	0.0	2261.9	3.1	0.2
	600	1319.7	1344.1	24.4	61.7	0.0	9894.3	34.4	2.0
	1000	1132.2	1149.7	17.5	54.9	0.0	19176.6	90.5	7.6

以上の結果から、次のことが分かる。

- 代理制約緩和法により、かなり精度の良い近似解が短時間で得られる。
- 目的関数値 (下界値 \underline{z} で代用) は n とともに減少し、 K の増加とともに増加する。これは、問題 MMAP の性格から予想されることである。
- n , K の増加にともなう、上下界値の計算時間 (CPU₁) が増大する。 δ の増加にともなう CPU₁ の増加はさほど顕著でない。
- 最適単体表の復元と釘付けに要する時間 (CPU₂) は n とともに増大するが、 K , δ の増加による影響はさほど大きくない。
- 釘付けテストにより、問題はきわめて大幅に縮小される。

4.3 XPRESS-MP による直接解法

K, δ, n の各値について例題をランダムに 10 題発生した。ここで、XPRESS-MP は DELL DIMENSION 8400 (Pentium(R)4 3.40GHz) 上で動かして実験を行う。計算はタイムリミットを CPU 時間 600 秒として打ち切ったが、図 1 には 600 秒以内に 10 例題がすべて解けたケースについてのみ、平均をグラフ化した。

$\delta = 0.3$ の場合でもシナリオ数が 8 以上となると XPRESS-MP では解き難い問題が多くなり、 $\delta = 0.6, \delta = 0.9$ ではシナリオ数 K 、商品数 n がかなり小さいケースでないと難しいことが分かる。

4.4 釘付けテスト経由の厳密解法

MMAP の厳密解法について数値実験を行う。実験計画は第 4 章と同様にそれぞれの δ に対し、 $n = 200 \sim 1000$ までの範囲で行い、各ケースについて 10 題をランダムに生成して平均値を計測した。解法としては上下界値を求め、第 3 章の（改良された）釘付けテストによって問題を縮小して XPRESS-MP で解く。この方法を以降「釘付け経由の厳密解法」と呼ぶ。

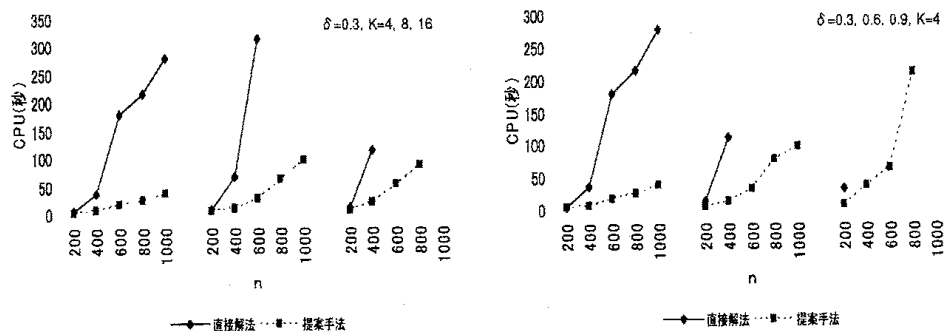


図 1: XPRESS-MP による直接解法と釘付けテスト経由の厳密解法

以上の結果から、次のことが分かる。

- 直接解法に比べ、釘付けテストを経由する方法は解ける問題のサイズ (K, n) がかなり大きくなり、同じサイズの問題では計算時間が数倍から数十倍速くなっている。
- 相関パラメータとシナリオ数が大きい場合は、釘付け経由の方法でもかなり問題が解き難くなる。

5. 結論

本研究では多目的割当問題のミニマックス最適化 (MMAP) を定式化し、その解法を検討した。主な成果は以下のとおりである。

- MMAP に代理制約緩和と劣勾配法（あるいは 2 分探索法）を適用し、上下界値を得るアルゴリズムを実現した。
- 代理制約緩和問題が（連続型の）割当問題となることから、最適単体表を復元することにより通常の 0-1 計画問題に適用される釘付けテストが MMAP にも適用可能であることを示した。
- n が大きい場合、普通の釘付けテストで計算すると膨大な計算時間がかかるので、それに対処するために改善釘付けテストを提案した。
- 種々のサイズとタイプの例題について数値実験を行い、上の方法の評価を行った。以上の結果、 n が 200 から 1000 までの MMAP が数分の計算時間で解けるようになった。

このように、釘付けテストにより一般に問題が大幅に縮小されるが、例えば $n = 1000$ でも変数 x_{ij} の数は 100 万個なので、このうち 99 万個が固定されたとしても、残った 1 万変数の問題を XPRESS-MP で解くことは困難である。そこで、今後の課題として、縮小された問題を商用ソルバーで解くのではなく、専用のアルゴリズムを構築して解くなどの対処法が考えられる。

参考文献

- [1] R. K. Ahuja, T. L. Magnanti, J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, Englewood Cliffs, 1993.
- [2] V. Chvátal, *Linear Programming*, Freeman and Company, San Francisco, 1983.
- [3] Z. Cimen, *A multi-objective decision support model for the Turkish armed forces personnel assignment system*, M.S. thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, 2001.
- [4] Fisher, M: The Lagrangian Relaxation Method for Solving Integer Programming
- [5] H. J. Greenberg, "Solving integer programming problems," *OR/MS Today* **26**(1999), 64-81.
- [6] M. R. Garey, D. S. Johnson: *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman and Company, San Francisco, 1979.
- [7] P. Kouvelis, G. Yu, *Robust Discrete Optimization and Its Applications*, Kluwer, Dordrecht, 1995.
- [8] H. B. Krass, "On the minimum cost flow problem," *Proc. American Math. Society*, **7**(1956), 48-50.
- [9] H. G. K. Z. "On the minimum cost flow problem," *Naval Research Logistics Quarterly*, **2**(1955), 83-95.
- [10] R. M. Nauss, *Parametric Integer Programming*, Univ. Missouri Press, Columbia, MI, 1979.
- [11] C. H. Papadimitriou, K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice Hall, Englewood Cliffs, 1982.
- [12] L. A. Wolsey, *Integer Programming*, John Wiley & Sons, New York, 1998.
- [13] "XPRESS-MP, <http://www.dashoptimization.com>, 2005.
- [14] 茨木俊秀, 組合せ最適化-分枝限定法を中心として, 産業図書, 1983.
- [15] 今野浩, 鈴木久敏 (編), 整数計画法と組合せ最適化, 日科技連, 1982.
- [16] 今野浩, 整数計画法, 産業図書, 1981.
- [17] 久保幹雄, 田村明久, 松井知己 (編), 応用数理計画ハンドブック, 朝倉書店, 2002.
- [18] 榊原 静, 中森真理雄, ベクトルコストによる割当て問題について, 日本 OR 学会春季研究発表会, 2-D-5, 東京農工大, H18. 3. 18.